Decision Tree Methods

CS 579: Machine Learning

Mayur Palankar

7th April 2005

Outline

- Introduction
- Univariate Decision Tree Algorithms
- Multivariate Decision Tree Algorithms
- Fisher's Linear Discriminant Analysis (LDA)
- Linear Discriminant Trees
- Comparison of Decision Tree Methods
- Conclusion
- Bibliography

INTRODUCTION

Decision Trees:

- Relatively simple knowledge formalism.
- Lacks expressive power of semantic networks (or other-first order representations).
- Learning Methodologies comparatively less complex.
- Capable of solving difficult problems of practical considerations.

Decision Tree Algorithms:

- Greedy algorithm.
- Composed of internal decision nodes and terminal leaves.
- Decision nodes implement Discriminant functions for classification.
- Value of the leaf node constitutes the output.

Introduction, cont…

Decision Trees Methods (based on Discriminant function):

1) Univariate Decision Trees (Simple but not flexible)
   - Only one feature (input dimension) is used leading to axis-aligned splits.
   - The feature can be continuous or discrete (m values) leading to binary or m-ary splits.

2) Linear Multivariate Trees (Flexible)
   - More than one feature is used leading to oblique splits.
   - Best linear combination of attributes to form an arbitrary hyper plane.
   - Exhaustive search (for thresholds or weights) not possible.
   - Most algorithms restricts nodes to binary splits.

3) Non-linear Multivariate Trees (Flexible but complex)
   - Able to make non-linear splits.

Univariate Decision Tree Algorithm

ID3 Algorithm:

- Simple Iterative Algorithm.
- No guarantee of optimal solution.
- Large trees and poor generalization.
- Best split selected on basis of Impurity measure (Entropy).
- Noise can cause attributes to become inadequate or give rise to complex decision trees.

Complexity:

At each node, its O (|C|.|A|), where |C| is the size of the training set and |A| is the number of attributes.

Total computation requirement is: O (|C|.|A|.|non-leaf nodes|).

Univariate Decision Tree Algorithm, cont…ID3 algorithm

Algorithm:

- A subset of the training set (called window) is chosen at random and a decision tree is formed as follows:
  - At a decision node, if all the objects are from the same class (or less then a threshold) then leaf is created labeled by majority class.
  - Else an *exhaustive* search is done for all possible splits and the best node is selected according to the impurity parameter. An attribute is chosen which gains the most information.
- If all other objects are correctly classified by the decision tree then the process terminates else the incorrectly classified objects are added to window and process continues.

(Since it is a divide and conquer strategy, it will always produce a decision tree that correctly classifies each object. However, the tree might not be optimal.)

Multivariate Decision Tree Algorithm

CART (Classification and Regression Tree):

The CART algorithm is used for finding the coefficients of the available features in a step-wise procedure.

- At each step, it cycles through the features x_1, x_2,…,x_n doing a search for an improved linear combination split.
  - Splits are compared using a chosen partition-merit criterion and the best split is used to update the weights of the linear combination split.
  - After the search is repeated for all features, the split which minimizes the impurity of the resulting partition is selected.
- The cycles end when the reduction of impurity is below a pre-specified constant.

Multivariate Decision Tree Algorithm, cont…

Neural Trees:

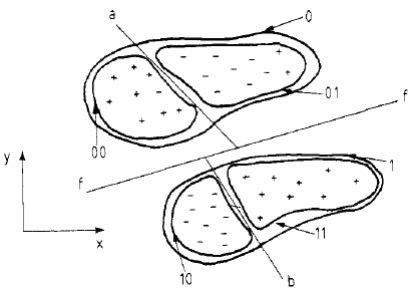
- The 'network' consists of perceptrons functionally organized in a binary tree ('neural tree').
- Inspired from a growth algorithm, the *tiling algorithm* (introduced for feed-forward neural networks).
- Convergence is guaranteed.
- Combines the advantage of a hierarchical organization and of the perceptron's ability to deal with many variables.
- Each neuron of the neural tree receives input from, and only from, the input layer and its output does not feed into any other neuron, but is used to propagate down a decision tree.
- This approach can be efficiently extended to classification in a multi-class problem.

Multivariate Decision Tree Algorithm, cont… Neural Trees

Algorithm (Classification with two classes):

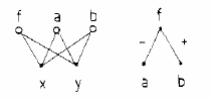
Consider a training set of 'patterns' belonging to two classes, A and B.

- All patterns (training set) are separated using *pocket algorithm*. This defines the first neuron and first binary classification: patterns divided into two subsets |0| and |1| with output -1 and +1 respectively.
  - The set |0| contains the ones 'recognized' as belonging to class A and set |1| to class B.
- If each subset '*faithful*', stop.
- If |0| is '*unfaithful*', a new neuron |0| is build (by pocket algorithm) to separate patterns (in |0|) to form subsets |00| and |11|.
- If |1| is unfaithful, same procedure to form |10| & |11|.
- This above procedure is repeated for all unfaithful subset till they all are faithful.



Multivariate Decision Tree Algorithm, cont… Neural Trees

The hierarchical organization put in memory in the form of a binary tree with each node associated with one neuron or a branch termination (assigned class numbers).



How the tree is used:

The pattern is presented to the first neuron and output from first neuron is read.

Based on -1 or +1, it looks at the output of neuron |0| or |1|.

If its 1, then it looks for the output of |10| or |11| based on the output of |1| and so on till one reaches a termination node.

The output is the class number.

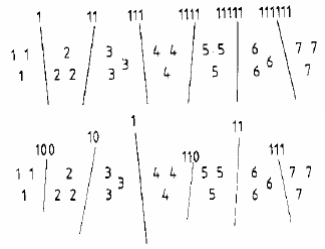
Learning schemes for single neuron:

Many Learning schemes can be considered. The pocket algorithm is as below:

- A standard perceptron algorithm (convergence algorithm) is run and the best '*coupling*' found is kept in the pocket (in memory).
- At each time, when the current coupling's are updated in the perceptron algorithm, the number of errors given by the current couplings are compared with the one in the pocket. The one with smaller errors is kept in the pocket.

Other criteria's:

- Information Theory.
- PCA (Principal Component Analysis).
  - Balanced partitions.



Non-optimal tree formation

PCA based strategy

Multivariate Decision Tree Algorithm, cont… Neural Trees

Generalization to Multi-class classification:

Consider K classes.

- K different binary classifications are done. i vs. (non i) where i = 1…K.
- Done by running K different pocket algorithms with the ith one trying to separate patterns from class i from all other patterns.
- According to the information criterion selected in the algorithm, the coupling's (given by the perceptrons) with the 'best score' is selected and subset is generated.
- The process is repeated till all subsets are 'faithful'.

Non-Linear Multivariate Decision Tree:

Non-linear multivariate decision trees can be generated by using multilayer perceptrons implementing non-linear boundary in input space.

Multivariate Decision Tree Algorithm, cont…

Linear Machine Decision Trees (LMDT):

1. If all the instances are from a single class, then set TREE to be a leaf node with the class name of the single class, return.
2. Otherwise, set TREE to be a decision node containing a test constructed by training a linear machine.
3. If the test partitions the instances into two or more subsets, then for each subset build a sub-tree recursively, return.
4. Otherwise, set TREE to be a leaf node with the class name of the most frequently occurring class, return.

- There is no heuristic measure (like information gain) but LMDT trains a linear machine which serves as a multivariate test at a decision node

Multivariate Decision Tree Algorithm, cont…LMDT

Linear Machine:

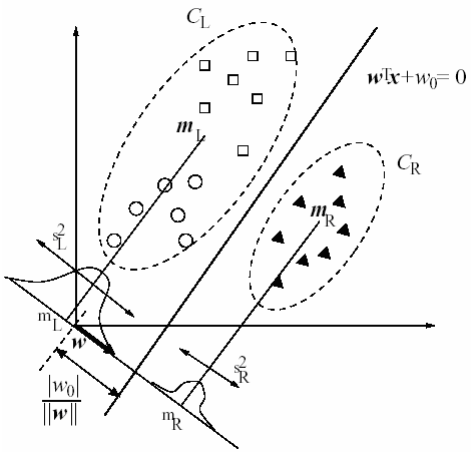
- It is a multi-class linear discriminant.
- It is a set of K linear discriminant functions that are collectively used to assign an instance to one of the K classes.
- Let $\boldsymbol{X}$ be an instance description. Then, each discriminant function $g_i(\boldsymbol{X})$, i = 1…K, has the form $\mathbf{W}^T_i\boldsymbol{X}$, where $\mathbf{W}$ is a vector of adjustable coefficients called weights.
- It infers $\boldsymbol{X}$ belongs to a class i iff (for all j, i ≠ j) $g_i(\boldsymbol{X}) > g_j(\boldsymbol{X})$.
- If there is no unique maximum, then classification is undefined.
- Weights are adjusted in response to misclassification of an instance. Weights of $\mathbf{W}_i$ are increased, where i is the class to which the instance belongs to and weights of $\mathbf{W}_j$ are decreased, where j is the class to which the linear machine erroneously classifies the instance.

Fisher's Linear Discriminant Analysis (LDA)

- Finds the best split given two distinct groups of classes.
- The two groups, C_L and C_R, are disjoint and can contain different input classes.
- Criterion:

  Maximize the following ratio

  $$\frac{\text{between-class variance}}{\text{with-in class variance}}$$



Fisher's Linear Discriminant Analysis (LDA), cont…

Find direction ***w*** to maximize J_F.

where, m_L and m_R are left and right group means and

S_w is within-class covariance matrix and

Σ_L and Σ_R are covariance matrix of C_L and C_R and

n_L and n_R are number of samples in each group.

$$J_F = \frac{|w^T(m_L - m_R)|^2}{|w^T \mathbf{S}_w w|}$$

$$m_L = \frac{1}{n_L} \sum_{x \in C_L} x$$

$$\mathbf{S}_w = \frac{1}{n_L + n_R}(n_L \mathbf{\Sigma}_L + n_R \mathbf{\Sigma}_R)$$

$$\mathbf{\Sigma}_L = \sum_{x \in C_L} (x - m_L)(x - m_L)^T$$

Solution for ***w*** that maximizes J_F can be obtained by differentiating it with respect to ***w*** and equating it to zero.

$$w = \mathbf{S}_w^{-1}(m_L - m_R)$$

$$w_0 = -\frac{1}{2}(m_L + m_R)^T \mathbf{S}_w^{-1}(m_L - m_R) - \log\frac{n_L}{n_R}$$

Linear Discriminant Trees

- Can perform classification for K classes (K > 2).
- Done as a nested optimization problem
  - In the inner optimization problem, Fisher's LDA finds the best split for given two groups of classes.
  - In the outer optimization problem, a method to find the best split of K classes into two groups.
    - Cannot test all possible partitions. For K > 2, the total possible partitions available are $2^{(K-1)} - 1$. So number of available partitions grows exponentially. The partition is found using local search method.

(This technique is very similar to Neural trees in terms of concept. It has no iterative training but (simple) calculation with matrices. So less training time is expected.)

Linear Discriminant Trees, cont…

Class Separation (Selection Method):

Let m be a decision node and C = $\{C_1,\ldots,C_K\}$ be the set of K classes at node m.

1. Select two classes C_i and C_j at random and put one in C^L_m and other in C^R_m. (Instead of random assignments, the classes with the largest inter-mean distance can be found and used.)
2. Train the discriminant with given partition using LDA. Don't consider other classes.
3. For other classes, find the class C_k that is best placed into one of the partitions as quantified by an impurity measure.
4. Add C_k to C^L_m or C^R_m depending on which side its instances fall more.

Repeat steps 2 to 4 till no more classes are left.

- This algorithm is sensitive to the initial class partition.
- The algorithm traces steps 2 to 4, (K-2) times. Its complexity is O(K).

Linear Discriminant Trees, cont…

Problem:

- If there is a linear dependency between two or more features then $\mathbf{S}_w$ becomes singular. So we cant find $\mathbf{S}_w^{-1}$.

Solution:

Principal Component Analysis (PCA)

- Find the eigenvectors and eigenvalues of the matrix $\mathbf{S}_w$, sort the eigenvalues and get rid of eigenvectors with small eigenvalues.
- For example, if eigenvalues are $\lambda_1,\ldots, \lambda_d$ (decreasing order). We find k eigenvectors that explain more than ϵ of the variance.

$$(\lambda_1 + \lambda_2 + \ldots + \lambda_k) / (\lambda_1 + \lambda_2 + \ldots + \lambda_k + \ldots + \lambda_d) > \epsilon$$

- The instances are then projected from the original d-dimensional space into the new k-dimensional space defined by the leading k eigenvectors. As eigenvectors are orthogonal, the new $\mathbf{S}_w$ has a inverse.
- PCA is a dimensionality reducing feature extraction process.

Comparison of Decision tree methods

Test Results:

- Trees developed with 5 % (training data) pre-pruning.
- Two-fold cross-validation runs on each data set.
- Ten runs were conducted reporting average and standard deviation on each data set.
- Testing criteria:
  - Tree size (Number of nodes) (Complexity not measured)
  - Accuracy
  - Learning time (seconds) (Pentium III-450)
- Neural Trees: Single layer perceptron (Linear multivariate).
- LDA Trees: PCA used with ϵ ranging from 0.9 to 0.99

  ϵ ↑ accuracy ↑ with tree size and learning time nearly constant.

Comparison of Decision tree methods, cont…

Description of the Data Sets:

SET	CLASSES	INSTANCES	FEATURES
CylINDER	2	541	36
DerMATOLOGY	6	366	35
EcoLI	8	336	8
FlaRE	3	323	11
Glass	7	214	10
HepATITIS	2	155	20
HorSE	2	368	27

- Data sets taken from UCI repository
- Cylinder, Hepatitis and Horse data sets have missing values.

Comparison of Decision tree methods, cont…

Tree Size:

Binary Splits

High Difference

SET	ID3	CART	ID-LP	LMDT	ID-LDA PCA WITH $\epsilon = 0.99$
CYL	54.6± 6.4	45.6± 4.9	8.4±1.9	23.8± 2.5	16.40± 8.95
DER	20.0± 3.0	28.0± 5.6	9.0±1.0	7.0± 0.0	12.80± 1.48
ECO	34.0± 3.0	34.0± 5.0	11.0±3.0	31.0± 5.0	20.00± 2.71
FLA	38.0± 4.7	33.8± 6.2	3.2±2.2	15.0± 9.3	5.60± 3.13
GLA	38.4± 5.8	42.4± 4.1	10.2±4.6	38.3±10.9	26.20± 4.44
HEP	19.6± 3.8	14.0± 3.4	3.0±0.0	10.6± 3.0	8.60± 3.10

Observation:

ID-LP < ID-LDA < LMDT < CART < ID3

Comparison of Decision tree methods, cont…

Accuracy:

SET	ID3	CART	ID-LP	LMDT	ID-LDA PCA WITH $\epsilon = 0.99$
CYL	68.7± 2.0	59.5±4.0	70.2± 4.5	68.6±2.9	69.80±3.01
DER	92.8± 2.4	80.9±4.6	85.7± 7.1	96.4±1.3	96.17±1.59
ECO	78.1± 3.6	74.6±3.8	82.6± 4.1	81.2±3.5	83.75±2.53
FLA	85.3± 2.0	81.5±3.6	88.4± 2.4	86.7±3.1	88.17±2.83
GLA	60.7± 6.2	53.9±4.2	55.0± 7.8	59.3±4.7	57.29±4.16
HEP	78.4± 3.7	79.0±4.0	84.1± 2.9	81.3±3.7	82.06±5.60

Pair wise comparison of accuracies

METHOD	ID3	CART	ID-LP	LMDT	ID-LDA
ID3		4	4	1	3
CART	0		2	1	1
ID-LP	4	6		0	1
LMDT	4	6	1		1
ID-LDA	5	8	2	2	

Observation:

ID-LP ≈ ID-LDA ≈ LMDT <$_{(?)}$ CART ≈$_{(?)}$ ID3

Comparison of Decision tree methods, cont…

Learning Time (seconds):

Very High

SET	ID3	CART	ID-LP	LMDT	ID-LDA PCA WITH $\epsilon = 0.99$
DER	6± 1	858± 170	42± 9	1± 0	16± 1
ECO	6± 1	221± 25	57± 15	1± 0	6± 1
FLA	5± 1	1032± 203	9± 4	1± 1	4± 3
GLA	7± 2	320± 25	33± 9	1± 0	7± 1
HEP	4± 1	209± 47	1± 0	0± 0	2± 1
HOR	9± 3	3481±1101	14± 2	4± 1	94± 40

Very Low

Observation:

LDMT < ID3 < ID-LDA < ID-LP < CART

Comparison of Decision tree methods, cont…

Accuracy vs. Tree size:

ID-LP: generates smallest and most accurate trees.
ID-LDA (very close), LMDT, ID3 and then CART.

Tree size vs. Learning time:

ID3 & LMDT: Fastest learners.
ID-LDA, ID-LP and then CART.

Tree size vs. Learning time:

ID3 & LMDT: Fast but generates large trees.
ID-LP & ID-LDA: Slow but generates small trees.
CART: Slow and generates large trees.

CONCLUSION

- When data set is small and has few classes:
  Univariate (ID3): Has better performance than multivariate linear methods with simple interpretable rules.

- When variables are highly correlated:
  Multivariate Method
  Linear Discriminant Trees (ID-LDA): (Learning time critical)
  ID-LDA > CART (accuracy, size & time)
  > LMDT (size) & ≈ LMDT (accuracy)
  > ID-LP (time) & ≈ ID-LP (accuracy)
  LMDT: Learning time ↓ but complexity ↑.
  Linear Perceptron Neural Trees (ID-LP): (Smallest Trees)
  ID-LP > ID-LDA (size)

Questions ??

Bibliography

- Induction of Decision Trees
  by J. R. Quinlan
- An Introduction to Classification and Regression Tree (CART) Analysis
  by Roger J. Lewis
- Linear Machine Decision Trees
  by Paul E. Utgoff and Carla E. Brodley
- Neural Trees: a new tool for classification
  by J. A. Sirat and J. P. Nadal
- Linear Discriminant Trees
  by Olcay Taner Yildiz and Ethem Alpaydin
- Introduction to Machine Learning
  by Ethem Alpaydin